VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8228

COMPARING OPTUNA AND HYPEROPT FOR MOBILENETV3 HYPERPARAMETER OPTIMIZATION IN CORN LEAF DISEASE CLASSIFICATION

Nur Rachmat[1*]; Anugerah Widi[2*]

Informatics[1]
Information Systems[2]
Universitas Multi Data Palembang, Palembang, Indonesia[1,2]
https://www.mdp.ac.id[1,2]
nur.rachmat@mdp.ac.id*, widi@mdp.ac.id

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *Corn leaf diseases pose a significant threat to agricultural productivity, necessitating accurate and efficient detection methods. While extensive deep learning research has been applied to plant disease classification, comparative studies of Hyperparameter Optimization (HPO) frameworks on lightweight architectures such as MobileNetV3 remain scarce. This study aims to compare the performance of Optuna and Hyperopt in optimizing the hyperparameters of MobileNetV3 for corn leaf disease classification. The Kaggle Corn Leaf Disease dataset, comprising 4,000 images across four classes (leaf blight, leaf spot, rust, and healthy), was split 80:10:10 for training, validation, and testing. The hyperparameter search space encompassed learning rate, optimizer type, batch size, and dropout rate, with each framework limited to 15 trials. Results indicate that for MobileNetV3-Small, Optuna achieved an accuracy of 96.75% (F1: 96.66%), marginally outperforming Hyperopt (96.50%; F1: 96.39%). Conversely, for MobileNetV3-Large, Hyperopt demonstrated superior performance with an accuracy of 97.00% (F1: 96.96%) compared to Optuna (96.25%; F1: 96.14%). The Wilcoxon signed-rank test revealed no statistically significant difference between the two frameworks (p > 0.05). These findings suggest that both frameworks are statistically equivalent, and MobileNetV3-Small offers the most favorable balance between classification accuracy and computational efficiency for mobile-based plant disease detection applications.*

Keywords: *Automated Hyperparameter Tuning; Bayesian Optimization; Corn Leaf Disease; Deep Learning; MobileNetV3.*

Intisari— *Penyakit daun jagung merupakan ancaman serius bagi produktivitas pertanian, sehingga membutuhkan metode deteksi yang akurat dan efisien. Meskipun deep learning telah banyak diterapkan untuk klasifikasi penyakit tanaman, studi yang secara spesifik membandingkan kinerja dua kerangka kerja Hyperparameter Optimization (HPO) pada arsitektur ringan seperti MobileNetV3 masih sangat terbatas. Penelitian ini bertujuan membandingkan kinerja Optuna dan Hyperopt dalam mengoptimalkan hiperparameter MobileNetV3 untuk klasifikasi penyakit daun jagung. Dataset Corn Leaf Disease dari Kaggle digunakan, yang terdiri dari 4.000 citra dalam empat kelas (leaf blight, leaf spot, rust, dan healthy), dengan pembagian data 80:10:10 untuk pelatihan, validasi, dan pengujian. Ruang pencarian hiperparameter mencakup learning rate, jenis optimizer, batch size, dan dropout rate, dengan setiap kerangka kerja dibatasi hingga 15 percobaan. Hasil penelitian menunjukkan bahwa pada arsitektur MobileNetV3-Small, Optuna mencapai akurasi 96,75% (F1: 96,66%), sedikit lebih tinggi dibandingkan Hyperopt (96,50%; F1: 96,39%). Sebaliknya, pada MobileNetV3-Large, Hyperopt lebih unggul dengan akurasi 97,00% (F1: 96,96%) dibandingkan Optuna (96,25%; F1: 96,14%). Uji Wilcoxon signed-rank menunjukkan bahwa tidak ditemukan perbedaan kinerja yang signifikan secara statistik antara kedua kerangka kerja (p > 0,05). Temuan ini mengindikasikan bahwa secara statistik, kedua kerangka kerja tersebut setara. Lebih lanjut, MobileNetV3-*

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8228

Small menawarkan keseimbangan terbaik antara akurasi dan efisiensi komputasi, menjadikannya pilihan ideal untuk aplikasi deteksi penyakit berbasis perangkat bergerak.

Kata Kunci: *Optimasi Hiperparameter Otomatis; Optimasi Bayesian; Penyakit Daun Jagung; Deep Learning; MobileNetV3.*

INTRODUCTION

Corn is a strategic food commodity crucial for food security and animal feed production in Indonesia [1], [2]. Corn production is significantly influenced by plant health; specifically, leaf diseases such as leaf spot, leaf blight, and rust pose significant threats that can substantially reduce crop yields [2], [3], [4], [5], [6]. Early identification of leaf diseases is essential for preventing disease proliferation and mitigating yield losses [2], [7], [8]. However, conventional methods based on visual observation by farmers or agricultural experts are often time-consuming, subjective, and prone to errors, particularly given the variability in field conditions, lighting, and environmental factors [7], [8]. The advancement of computer vision technology based on deep learning has created significant opportunities for the development of rapid and accurate automated classification systems deployable on mobile devices [9], [10], [11].

Numerous studies have utilized Convolutional Neural Network (CNN) architectures for corn leaf disease classification [9], [10], [12], [13], [14]. For example, research employing the MobileNetV3 architecture achieved high accuracies: the MobileNetV3-Small variant reached 99.5%, and the MobileNetV3-Large variant achieved 99.0% in corn disease classification, both after hyperparameter tuning on learning rate and batch size [15]. Despite these remarkably high results, hyperparameter tuning is a time-consuming process that does not guarantee optimal configuration [16]. Furthermore, manual tuning processes often depend on researcher experience and trial-and-error, making them inefficient, especially for complex deep learning models that require substantial computational resources [17].

Hyperparameter Optimization (HPO) is essential for improving machine learning model performance because hyperparameters significantly influence prediction quality. Conventional methods such as grid search and random search become inefficient at scale, prompting the development of more adaptive approaches like Bayesian optimization. Comparative studies demonstrate that Optuna and Hyperopt are effective HPO libraries; Optuna excels in algorithm selection problems, while Hyperopt is more optimal for MLP architecture optimization [18]. The application of Bayesian Optimization successfully improved ResNet-50 accuracy from 82.82% to 87.79% [19]. Another study achieved a 97% accuracy on SqueezeNet [20].

Despite extensive research on hyperparameter optimization in deep learning [21] [22], [23], [24], limited studies have specifically compared Optuna and Hyperopt on lightweight architectures such as MobileNetV3-Small for agricultural disease detection. Conventional approaches like random and grid search are less efficient at scale [21], [22]. This motivates the adoption of Bayesian-based frameworks such as Optuna [25] and Hyperopt, which leverage probabilistic surrogate models to more effectively navigate large hyperparameter search spaces [24]. Previous research focused on larger architectures (ResNet-50, AlexNet) or employed single HPO methods without comparative analysis. The novelty of this research lies in: (1) providing a comprehensive comparative analysis of two prominent Bayesian-based HPO frameworks (Optuna and Hyperopt) specifically on the MobileNetV3 architecture, which is optimized for mobile deployment; (2) evaluating multiple performance metrics including validation performance metrics, including validation accuracy, loss convergence, and computational efficiency, to determine the most suitable HPO framework; and (3) establishing empirical guidelines for selecting HPO frameworks in developing lightweight deep learning models for edge device deployment in agricultural applications.

Therefore, this study aims to compare Optuna and Hyperopt, two hyperparameter optimization methods, on MobileNetV3 architecture for corn leaf disease classification. Evaluation is conducted based on accuracy, convergence speed, and computational efficiency to determine the more appropriate method for developing lightweight and efficient classification models for devices with computational limitations.

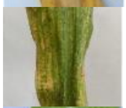
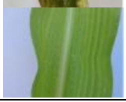
MATERIALS AND METHODS

Data Collection

This study utilizes the Corn Leaf Disease Dataset from the public Kaggle repository. The dataset comprises four classes of corn leaf images:

leaf spot, leaf blight, rust, and healthy. Each class contains 1,000 images, totaling 4,000 images for this research. The dataset distribution is presented in Table 1.

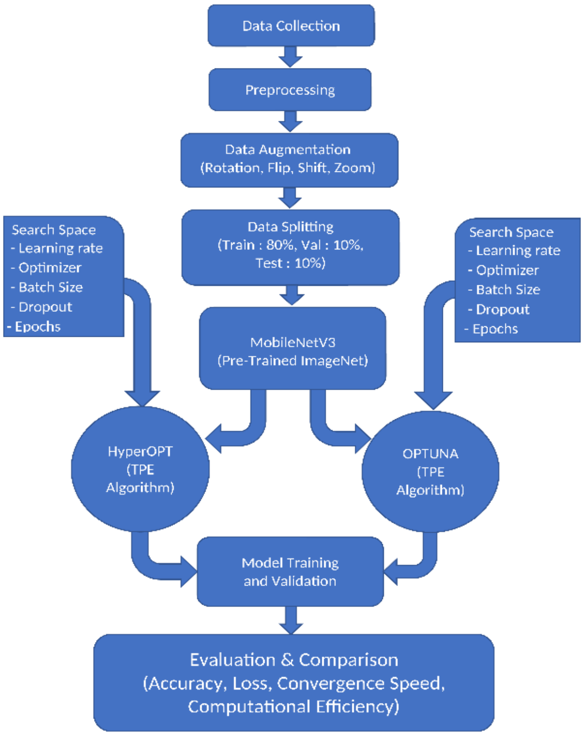
Table 1. Dataset Distribution

No	Class	Number of Images	Examples of Images
1	Leaf Spot (Bercak daun)	1.000	
2	Leaf Blight (Hawar daun)	1.000	
3	Rust (Karat daun)	1.000	
4	Healthy (Daun sehat)	1.000	

Source: (Research Results, 2025)

Research Framework

This research aims to develop a corn leaf disease classification model employing an optimized MobileNetV3 architecture. All stages of model development were systematically designed in accordance with deep learning best practices. The model development workflow is illustrated in Figure 1.



Source: (Research Results, 2025)
Figure 1. Research Framework

Data Processing

After collecting the dataset from public sources, a preprocessing stage was conducted to prepare the data for model training. This phase involved resizing images to 224×224 pixels and normalizing pixel values to the range [0,1] to accelerate model convergence.

Data Augmentation

Data augmentation techniques were applied to the training data, including random rotation, horizontal and vertical flipping, width and height shifting, and zooming to increase data diversity and reduce overfitting risk.

Data Splitting

The dataset was then randomly divided into three subsets: 80% for training, 10% for validation, and 10% for testing. The detailed data split is presented in Table 2.

Table 2. Image Data Splitting

No	Data Splitting	Class	Amount
1	Training	Leaf spot	800
2		Leaf Blight	800
3		Rust	800
4		Healthy	800
5	Validation	Leaf spot	100
6		Leaf Blight	100
7		Rust	100
8		Healthy	100
9	Testing	Leaf spot	100
10		Leaf Blight	100
11		Rust	100
12		Healthy	100

Source: (Research Results, 2025)

Table 2 shows the dataset distribution across four classes: leaf spots, leaf blight, rust, and healthy. The training set consists of 800 images per class (3,200 total), while validation and testing sets each contain 100 images per class (400 total each). The entire dataset comprises 4,000 images. This balanced distribution ensures representative training, validation, and testing results, avoiding class bias.

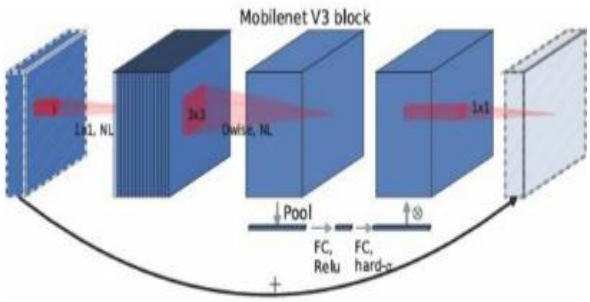
MobileNetV3 Architecture

MobileNetV3, the latest advancement in the MobileNet series, is meticulously designed for enhanced efficiency on mobile devices. Its development combines Neural Architecture Search (NAS), NetAdapt, and novel architectural innovations. This architecture is offered in two variants, MobileNetV3-Large and MobileNetV3-Small, both demonstrating improved accuracy and speed in classification, object detection, and semantic segmentation tasks compared to MobileNetV2 [26].

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8228

MobileNetV3 uses Inverted Residual Blocks, Squeeze-and-Excitation (SE) Modules, and Hard-Swish activation functions for efficient feature learning and computational lightness. This architecture is specifically optimized for mobile CPUs, delivering high performance with lower power consumption [26]. Figure 2 shows the MobileNetV3 block.



Source: (Howard [26], 2019)
Figure 2. MobileNetV3 Architecture

The Hard-Swish activation function is defined as follows:

$$h - swish(x) = x \cdot ReLU6(x + 3) / 6 \quad (1)$$

where ReLU6 is the rectified linear unit capped at 6, which provides a computationally efficient approximation of the Swish activation function suitable for mobile deployment.

Loss Function

The model employs categorical cross-entropy as the loss function for multi-class classification. The cross-entropy loss is calculated as:

$$L = -\sum_{c=1}^{C} y_c \log(p_c) \quad (2)$$

where C represents the number of classes (C=4 in this study), yc is the binary indicator (0 or 1) for class c, and pc is the predicted probability for class c.

Hyperparameter Tuning

Hyperparameter tuning for the MobileNetV3 architecture was conducted separately between Hyperopt and Optuna. Both frameworks utilize the Tree-structured Parzen Estimator (TPE) algorithm for Bayesian optimization. The hyperparameter search space is presented in Table 3.

Table 3. Hyperparameter Search Space

Hyperparameter	Search Range
Learning Rate	0.00001 – 0.01 (log-uniform)
Optimizer	Adam, SGD
Batch Size	16, 32, 64
Dropout Rate	0.2 – 0.5
Epochs	5 – 20
Maximum Trials	15

Source: (Research Results, 2025)

Model training was conducted on a local workstation running Windows 10, equipped with an Intel® Core™ i5-4210U processor (1.70 GHz, 4 CPUs, ~2.4 GHz) and 16 GB of RAM. The number of training epochs was included in the hyperparameter search space (range 5 to 20) and was therefore optimised jointly with the other hyperparameters, during each trial the model state achieving the highest validation accuracy was retained for evaluation. Model performance was evaluated using the accuracy metric, defined as:
Accuracy = (TP+TN)/(TP+TN+FP+FN) × 100% (3)
where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative.

RESULTS AND DISCUSSION

A comprehensive experimental study compared the performance of two hyperparameter optimization (HPO) frameworks, Optuna and HyperOpt, applied to MobileNetV3-Small and MobileNetV3-Large architectures for corn leaf disease classification. Each framework was allocated 15 optimization trials to explore the hyperparameter search space. Final performance metrics, including accuracy and F1-score, were derived from 5 independent experimental runs to ensure statistical robustness. Computational efficiency was quantified as the total optimization time (seconds) recorded for each framework and architecture. This rigorous design enables valid statistical comparison between the optimization frameworks and provides measurable insights into their computational efficiency.

MobileNetV3-Small Architecture Results

Table 4 presents the hyperparameter optimization results for MobileNetV3-Small, obtained using the HyperOpt framework across 15 trials. Validation accuracy varied substantially, ranging from 35.25% (Trial 9) to 97.75% (Trial 3), with a mean of 83.55% and a standard deviation of 19.31%, reflecting high sensitivity to hyperparameter initialization. The best performance was achieved in Trial 3 with 97.75% validation accuracy, followed closely by Trial 15 (97.50%) and Trial 2 (97.00%). These topperforming trials consistently employed lower learning rates (0.0001–0.001) and the Adam optimizer. This suggests that adaptive gradient-based optimization with conservative learning rates contributes to stable convergence. Conversely, trials exhibiting poor convergence, such as Trial 9 (35.25%) and Trial 11 (51.75%), were associated with high learning rates and SGD optimizer configurations, which are more susceptible to unstable gradient updates without momentum tuning. The high standard deviation further indicates that HyperOpt's TPE-based sampling initially explores a broad search space before narrowing toward promising regions, thus resulting in highly variable early trial outcomes.

Table 4. HyperOpt Optimization Results on MobileNetV3-Small

Trial	Val. Acc (%)
1	90.75
2	97.00
3	**97.75**
4	94.25
5	95.75
6	96.00
7	70.25
8	95.75
9	35.25
10	97.00
11	51.75
12	90.50
13	72.25
14	71.50
15	97.50

Source: (Research Results, 2025)

Optuna achieved superior optimization efficiency on MobileNetV3-Small, reaching a best validation accuracy of 99.00% in Trial 11. Its Tree-structured Parzen Estimator (TPE) algorithm effectively explored the hyperparameter search space, with most trials surpassing 90% validation accuracy. The optimization completed in 131,618 seconds (36.56 hours), a 24.4% reduction from HyperOpt's 174,131 seconds (48.37 hours).

MobileNetV3-Large Architecture Results

To provide a more comprehensive architectural comparison, experiments were extended to MobileNetV3-Large. Table 5 presents the HyperOpt optimization results across 15 trials for this architecture. The larger model demonstrated notably more consistent performance than MobileNetV3-Small, with validation accuracy ranging from 51.75% (Trial 13) to 97.75% (Trial 14). This reduced variance across trials suggests that MobileNetV3-Large's increased parameter capacity facilitates more stable training dynamics. Separately, Optuna achieved a best validation accuracy of 98.25% in Trial 5 for MobileNetV3-Large. Regarding computational efficiency, HyperOpt completed optimization for the larger architecture in 124,475 seconds (approximately 34.58 hours), demonstrating a 5.1% time advantage over Optuna, which required 131,199 seconds (approximately 36.44 hours).

It is important to note that all experiments were conducted in a CPU-only environment (Intel Core i5-4210U, 16 GB RAM) without GPU acceleration. Therefore, total optimization time served as the primary proxy for computational efficiency, as dedicated resource utilization monitoring tools were not employed. This constraint is acknowledged as a limitation, and future work is recommended to incorporate resource utilization metrics, such as CPU load and memory consumption, to provide a more comprehensive efficiency analysis.

Table 5. HyperOpt Optimization Results on MobileNetV3-Large

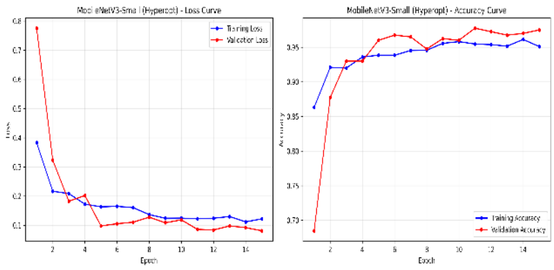
Trial	Val. Acc (%)
1	96.75
2	94.75
3	97.50
4	96.25
5	93.75
6	97.25
7	94.75
8	97.25
9	97.00
10	96.50
11	96.50
12	93.25
13	51.75
14	**97.75**
15	93.75

Source: (Research Results, 2025)

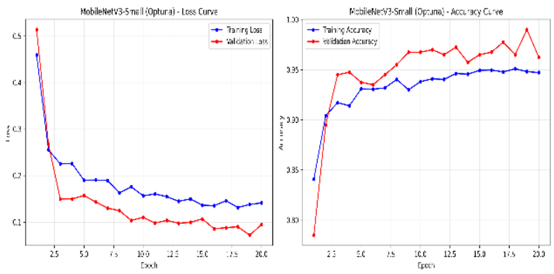
Training Dynamics Analysis

Figures 3 and 4 present the training dynamics of the best trial identified by each framework on MobileNetV3-Small. Optuna's best configuration reached 99.00% validation accuracy over 20 epochs, compared with 97.75% over 15 epochs for HyperOpt, the differing epoch counts arise because the number of epochs was itself part of the search space and was therefore selected independently by each framework. Both frameworks converged to comparable final validation accuracy, while Optuna completed the overall optimization 24.4% faster in wall-clock time.

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)
VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8228

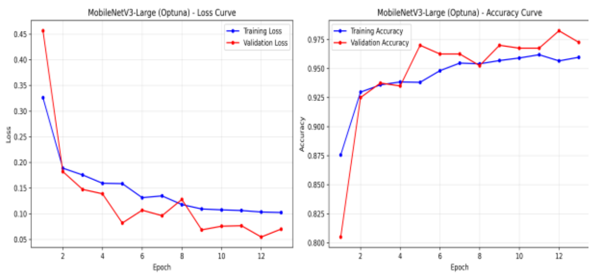


Source : (Reseach Results, 2025)
Figure 3. Loss and Accuracy Curve MobileNetV3-Small with Hyperopt



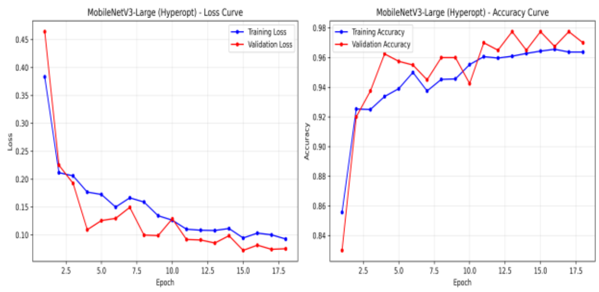
Source : (Reseach Results, 2025)
Figure 4. Loss and Accuracy Curve MobileNetV3-Small with Optuna

For deeper insight into the optimization behavior, Figures 5 and 6 present the training and validation curves for the best hyperparameter configurations identified by Optuna and HyperOpt, respectively, on the MobileNetV3-Large architecture. These curves illustrate the patterns of loss reduction and accuracy improvement throughout the training process. Figure 5 demonstrates the training dynamics of MobileNetV3-Large, optimized using Optuna. The loss curve rapidly descends from approximately 0.45 to below 0.20 within the first two epochs, then gradually converges to approximately 0.05-0.10 by epoch 12.



Source : (Reseach Results, 2025)
Figure 5 Training and Validation Curves for MobileNetV3-Large with Optuna Optimization

Figure 6 illustrates the training progression for MobileNetV3-Large with HyperOpt-optimized hyperparameters. The loss curve exhibited an initial rapid descent, decreasing from approximately 0.45 to 0.20 within the first three epochs, a rate comparable to Optuna. However, HyperOpt required approximately 18 epochs for final convergence, representing a 50% increase compared to Optuna's 12 epochs. The final validation loss for HyperOpt stabilized at approximately 0.08, which is slightly higher than Optuna's range of 0.05–0.10. The accuracy curve displayed consistent improvement, rising from approximately 83% to over 97%. Noticeable fluctuations in validation accuracy were observed between epochs 7 and 12. These oscillations, less prominent in the Optuna configuration, suggest that HyperOpt's selected learning rate might be slightly higher, leading to more volatile gradient updates during the mid-training phase. Despite HyperOpt's slower convergence, both frameworks ultimately achieved comparable final validation accuracy, indicating a marginal performance gap between the two methods.



Source : (Reseach Results, 2025)
Figure 6 Training and Validation Curves for MobileNetV3-Large with HyperOpt Optimization

Test Set Evaluation and Statistical Analysis

For robust generalization assessment, the optimal hyperparameter configurations identified by each framework were evaluated on the held-out test set over 5 independent experimental runs. Table 6 compares test set performance metrics, including accuracy and F1-score, and provides Wilcoxon signed-rank test p-values for statistical significance.

Table 6. Comprehensive Performance Comparison on Test Set

Configuration	Test Acc (%)	F1 (%)
MobileNetV3-Small		
Optuna	**96.75**	**96.66**
HyperOpt	96.50	96.39
p-value (Wilcoxon)	0.2733 (ns)	
MobileNetV3-Large		
Optuna	96.25	96.14
HyperOpt	**97.00**	**96.96**
p-value (Wilcoxon)	0.0656 (ns)	

Source : (Reseach Results, 2025)
Note: ns = not significant at α = 0.05; bold = best per architecture

Table 7. Computational Efficiency Comparison

Architecture	Framework	Time (h)
MobileNetV3-Small	Optuna	**36.56**
	HyperOpt	48.37
MobileNetV3-Large	Optuna	36.44
	HyperOpt	**34.58**

Source : (Reseach Results, 2025)
Note: bold = faster optimization time per architecture

The Wilcoxon signed-rank test assessed the statistical significance of performance differences between Optuna and HyperOpt across the 5 experimental runs. For MobileNetV3-Small, a p-value of 0.2733 ($p > 0.05$) indicated no significant difference between frameworks, despite Optuna's marginally higher accuracy (96.75% vs 96.50%) and F1-score (96.66% vs 96.39%). Similarly, for MobileNetV3-Large, a p-value of 0.0656 ($p > 0.05$) indicates that the observed performance difference, with HyperOpt achieving 97.00% compared to Optuna's 96.25%, is not statistically significant at the conventional $\alpha = 0.05$ threshold. Nevertheless, the result approaches marginal significance.

CONCLUSION

This study compared Optuna and HyperOpt for automated hyperparameter optimization on MobileNetV3 architectures in corn leaf disease classification. The Wilcoxon signed-rank test confirmed no significant performance difference between the frameworks across architectures, establishing both as equally viable. However, optimization efficiency was architecture-dependent: Optuna completed the optimization faster on the smaller architecture, while HyperOpt demonstrated a time advantage on the larger model, suggesting sampling strategy effectiveness scales with model complexity. This study provides empirical evidence that the choice between Optuna and HyperOpt should be guided by architectural considerations rather than expected accuracy gains. MobileNetV3-Small is recommended for deployment in mobile and edge-based agricultural systems, offering competitive classification performance and lower computational overhead. Future work should extend this comparison to a broader range of architectures, incorporate GPU-based resource monitoring, and explore additional HPO strategies such as population-based training to advance lightweight model development for precision agriculture.

REFERENCE

[1] R. D. Yofa, Sumedi, and S. H. Susilowati, “Dinamika perubahan nilai indikator strategis pembangunan pertanian dan transformasi perdesaan,” *Anal. Kebijak. Pertan.*, vol. 22, no. 1, pp. 51–62, 2024, doi: 10.21082/akp.v22i1.51-62.

[2] M. Lihawa *et al.*, “Pengembangan Aplikasi Sistem Pakar Deteksi Dini Hama dan Penyakit Tanaman Jagung,” *J. Penelit. Pertan. Terap.*, vol. 24, no. 1, pp. 58–66, 2024, doi: 10.25181/jppt.v24i1.3039.

[3] Reymas M.R. Ruimassa, Rosdiana Sari, and Eko Agus Martanto, “Interaksi Faktor Iklim dan Varietas terhadap Laju Perkembangan Penyakit Karat Daun (Puccinia polysora Undrew) pada Jagung (Zea mays L.),” *J. Trit.*, vol. 14, no. 1, pp. 141–152, 2023, doi: 10.47687/jt.v14i1.371.

[4] D. Ramadona *et al.*, “Infentarization [sic] of leaf blight in maize in Suka Menang Village, Gelumbang Subdistrict, Muara Enim,” *Penerbit Percetakan Univ. Sriwij.*, vol. 6051, pp. 237–251, 2023, [Online]. Available: https://conference.unsri.ac.id/index.php/lahansuboptimal/article/viewFile/2749/1659

[5] G. Shandilya, S. Gupta, H. G. Mohamed, S. Bharany, A. U. Rehman, and S. Hussen, “Enhanced Maize Leaf Disease Detection and Classification Using an Integrated CNN-ViT Model,” *Food Sci. Nutr.*, vol. 13, no. 7, p. e70513, 2025, doi: 10.1002/fsn3.70513.

[6] S. Sulastri, T. Surjana, and M. Syafi’i, “Intensitas Serangan Penyakit Penting pada Beberapa Galur Jagung Manis (Zea mays Var. saccharata Sturt) MS-UNSIKA Generasi M8,” *J. Agroplasma*, vol. 11, no. 2, pp. 375–381, 2024, doi: 10.36987/agroplasma.v11i2.5944.

[7] A. Pfordt and S. Paulus, “A review on detection and differentiation of maize diseases and pests by imaging sensors,” *J. Plant Dis. Prot.*, vol. 132, no. 40, pp. 1–21, 2025, doi: 10.1007/s41348-024-01019-4.

[8] A. Dolatabadian, T. X. Neik, M. F. Danilevicz, S. R. Upadhyaya, J. Batley, and D. Edwards, “Image-based crop disease detection using machine learning,” *Plant Pathol.*, vol. 74, no. 1, pp. 18–38, 2025, doi: 10.1111/ppa.14006.

[9] F. Khan, N. Zafar, M. N. Tahir, M. Aqib, H. Waheed, and Z. Haroon, “A mobile-based system for maize plant leaf disease detection and classification using deep learning,” *Front. Plant Sci.*, vol. 14, no. May, pp. 1–18, 2023, doi: 10.3389/fpls.2023.1079366.

[10] G. T. Askale, A. B. Yibel, B. M. Taye, and G. D. Wubneh, “Mobile based deep CNN model for maize leaf disease detection and

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8228

classification," *Plant Methods*, vol. 21, no. 1, 2025, doi: 10.1186/s13007-025-01386-5.

[11] J. Zhang, X. Yang, X. Fu, B. Wang, and H. Li, "LDL-MobileNetV3S: an enhanced lightweight MobileNetV3-small model for potato leaf disease diagnosis through multi-module fusion," *Front. Plant Sci.*, vol. 16, p. 1656731, 2025, doi: 10.3389/fpls.2025.1656731.

[12] S. S. Putro, N. Ansori, M. Fuad, E. M. S. Rochman, Y. P. Asmara, and A. Rachmad, "Corn Leaf Disease Classification Using Convolutional Neural Network Based on MobileNetV2 with RMSProp Optimization," *Math. Model. Eng. Probl.*, vol. 12, no. 2, pp. 465–474, 2025, doi: 10.18280/mmep.120211.

[13] W. G. Pamungkas, M. I. P. Wardhana, Z. Sari, and Y. Azhar, "Leaf Image Identification: CNN with EfficientNet-B0 and ResNet-50 Used to Classified Corn Disease," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 2, pp. 326–333, 2023, doi: 10.29207/resti.v7i2.4736.

[14] S. Timilsina, S. Sharma, and S. Kondo, "Advancements in maize leaf disease detection, segmentation and classification: A review," *Biosyst. Eng.*, vol. 255, p. 104162, 2025, doi: 10.1016/j.biosystemseng.2025.104162.

[15] W. Maximilliano and N. Rachmat, "Comparative Analysis of MobileNetV3-Large and Small for Corn Leaf Disease Classification," *Brill. Res. Artif. Intell.*, vol. 5, no. 1, pp. 325–332, 2025, doi: 10.47709/brilliance.v5i1.6259.

[16] J. Won, H.-S. Lee, and J.-W. Lee, "A review on multi-fidelity hyperparameter optimization in machine learning," *ICT Express*, vol. 11, no. 2, pp. 245–257, 2025, doi: 10.1016/j.icte.2025.02.008.

[17] A. Morales-Hernández, I. Van Nieuwenhuyse, and S. Rojas Gonzalez, "A survey on multi-objective hyperparameter optimization algorithms for machine learning," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8043–8093, 2023, doi: 10.1007/s10462-022-10359-2.

[18] S. Shekhar, A. Bansode, and A. Salim, "A Comparative study of Hyper-Parameter Optimization Tools," *2021 IEEE Asia-Pacific Conf. Comput. Sci. Data Eng.*, pp. 1–6, 2021, doi: 10.1109/CSDE53843.2021.9718485.

[19] Y. A. Abdillah and K. Kusrini, "Corn Leaf Disease Classification Optimization Using Resnet50 Architecture Utilizing Bayesian Optimization," *J. Electr. Eng. Comput.*, vol. 7, no. 1, pp. 8–15, 2025, doi: 10.33650/jeecom.v7i1.9809.

[20] P. Theerthagiri, A. U. Ruby, J. G. C. Chandran, T. H. Sardar, and A. Shafeeq B. M., "Deep SqueezeNet learning model for diagnosis and prediction of maize leaf diseases," J. Big Data, vol. 11, no. 1, pp. 1–16, 2024, doi: 10.1186/s40537-024-00972-z.

[21] J. Bergstra, R. Bardenet, Y. Bengio, and B. K´egl, "Algorithms for Hyper-Parameter Optimization," *Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf

[22] J. Bergstra, D. Yamins, and D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures," *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28, no. 1, pp. 115–123, 2013, [Online]. Available: https://proceedings.mlr.press/v28/bergstra13.html

[23] Y. A. Ali, E. M. Awwad, M. Al-Razgan, and A. Maarouf, "Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity," *Processes*, vol. 11, no. 2, p. 349, 2023, doi: 10.3390/pr11020349.

[24] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges, " *WIREs Data Min. Knowl. Discov.*, vol. 13, no. 2, p. e1484, 2023, doi: 10.1002/widm.1484.

[25] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Appl. Data Sci. Track Pap.*, pp. 2623–2631, 2019, doi: 10.1145/3292500.3330701.

[26] A. Howard *et al.*, "Searching for MobileNetV3," *2019 IEEE/CVF Int. Conf. Comput. Vis.*, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.